#SHAREorg

SHARE
Technology · Connections · Results

Running Linux virtual servers on your XIV under z/VM

Gregory Powiedziuk
Transzap, Inc.

4:30 PM on Monday, Feb 4, 2013
Session 12955

http://www.linkedin.com/pub/grzegorz-powiedziuk/6/137/11a/

SHARE in San Francisco 2013

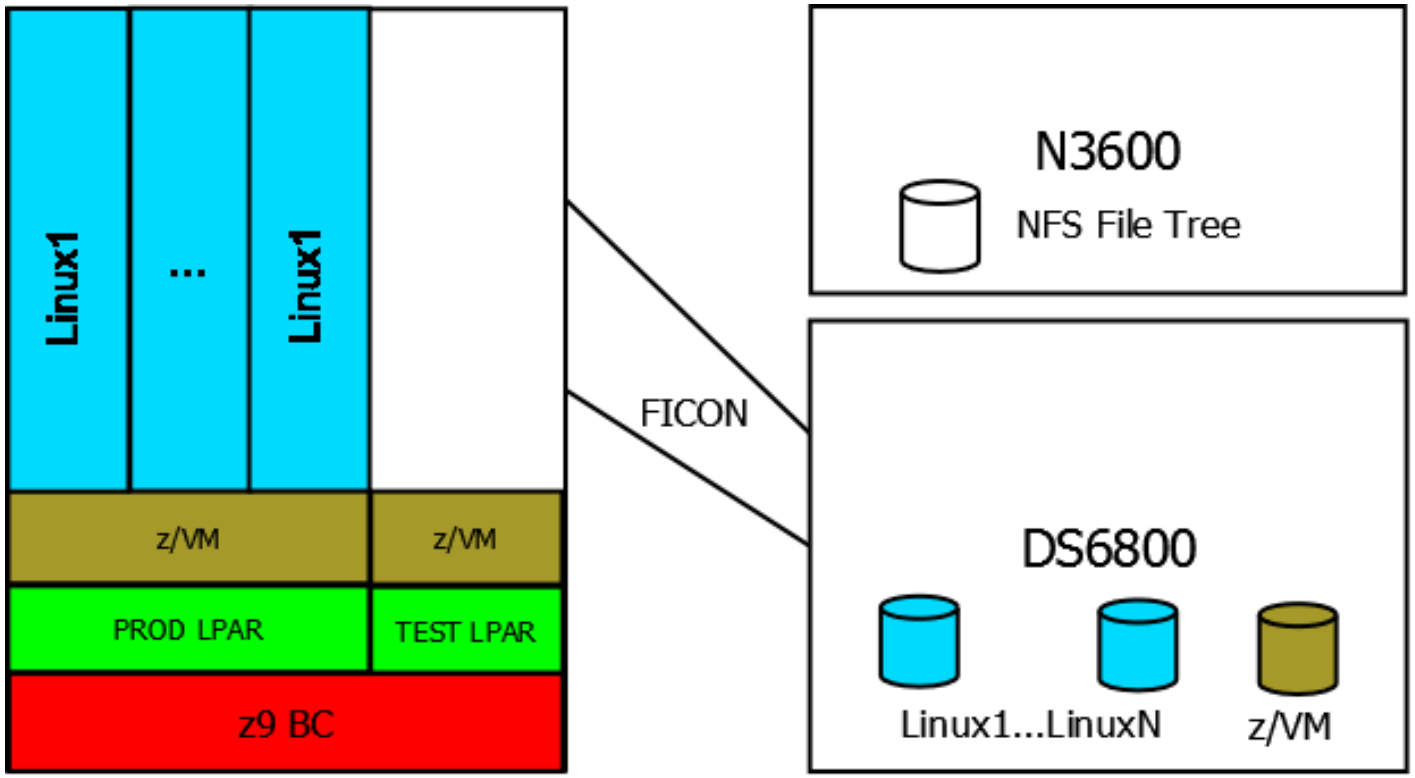
Table of contents

Who is Transzap?

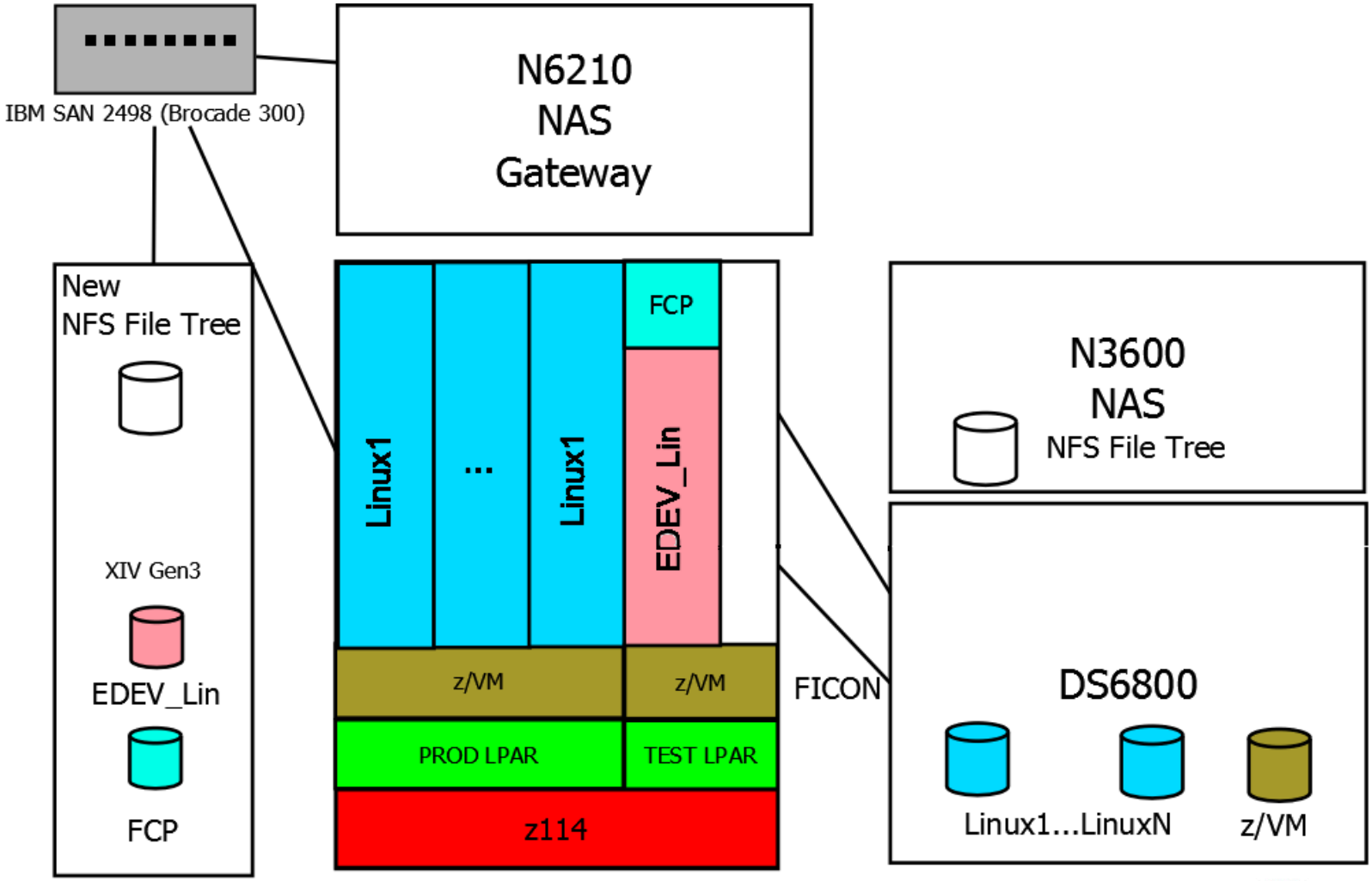
- Transzap, the parent company of Oildex, provides software solutions that significantly improve the flow of business information as well as operational efficiencies for companies in the North American oil and natural gas industry. Through its suite of energy industry software-as-a-service (SaaS) solutions, Oildex enables companies to collect, distribute, manage, and analyze essential business data - far more efficiently and faster than traditional desktop software systems.

Old setup

- Z9 BC
  - 5xIFL
  - 24GB
- N3600
- DS6800



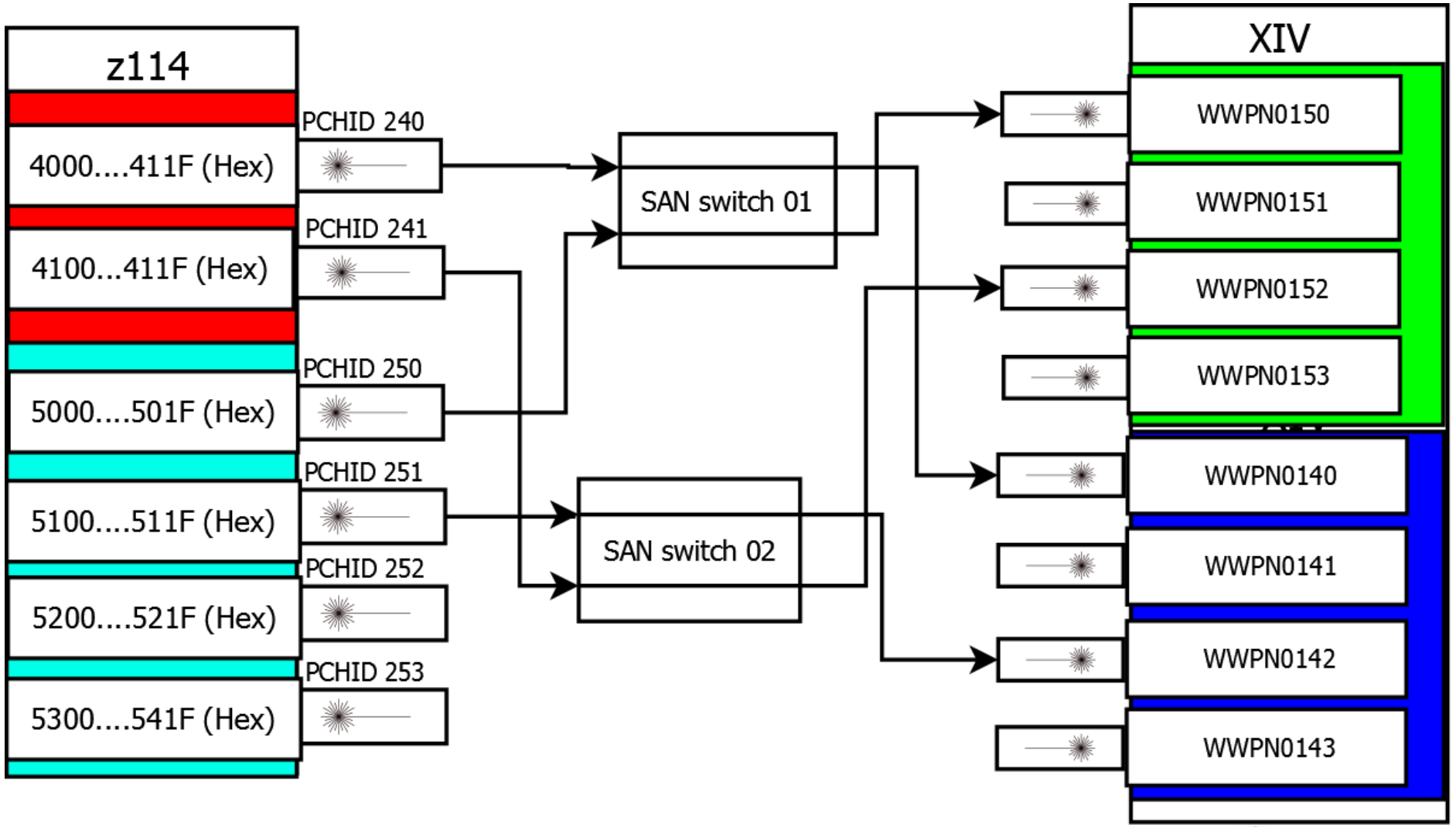
New setup



New setup

- z114 BC
  - 40GB Storage
  - 5xIFL
  - 6xFICON Express4 SX (FC) - FCP mode
  - 4xFICON Express4 LX (FICON) – FICON mode
- DS6800
- XIV Gen3 55TB
  - 6 Modules
  - 8xFC Ports 8Gbit
- IBM N6210 Gateway
  - No disks
  - 2xNetapp Controllers
  - 12xEth Ports1Gbit
  - 4xFC Ports 4Gbit
- 2xIBM SAN 2498

Multipath z114 -> SAN -> XIV



IOCP configuration

```
      CHPID PATH=(CSS(0),40),SHARED,
            PARTITION=((VMPR,VMTS,VMWEB),(=)),
            PCHID=240,TYPE=FCP
        CHPID PATH=(CSS(0),41),
            PARTITION=((VMPR,VMTS,VMWEB),(=)),
            PCHID=241,TYPE=FCP
        CHPID PATH=(CSS(0),50),SHARED,
            PARTITION=((VMPR,VMTS,VMWEB),(=)),
            PCHID=250,TYPE=FCP
        CHPID PATH=(CSS(0),51),
            PARTITION=((VMPR,VMTS,VMWEB),(=)),
            PCHID=251,TYPE=FCP
    CNTLUNIT CUNUMBR=040,PATH=((CSS(0),40)),UNIT=FCP
        IODEVICE ADDRESS=(4000,32),CUNUMBR=(040),UNIT=FCP
    CNTLUNIT CUNUMBR=041,PATH=((CSS(0),41)),UNIT=FCP
        IODEVICE ADDRESS=(4100,32),CUNUMBR=(041),UNIT=FCP
    CNTLUNIT CUNUMBR=050,PATH=((CSS(0),50)),UNIT=FCP
        IODEVICE ADDRESS=(5000,32),CUNUMBR=(050),UNIT=FCP
    CNTLUNIT CUNUMBR=051,PATH=((CSS(0),51)),UNIT=FCP
        IODEVICE ADDRESS=(5100,32),CUNUMBR=(051),UNIT=FCP
```

IOCP configuration

In above example:

- Set PCHID 240 to TYPE=FCP
- Set IODEVICE address to (4000,32)
- Set UNIT=FCP

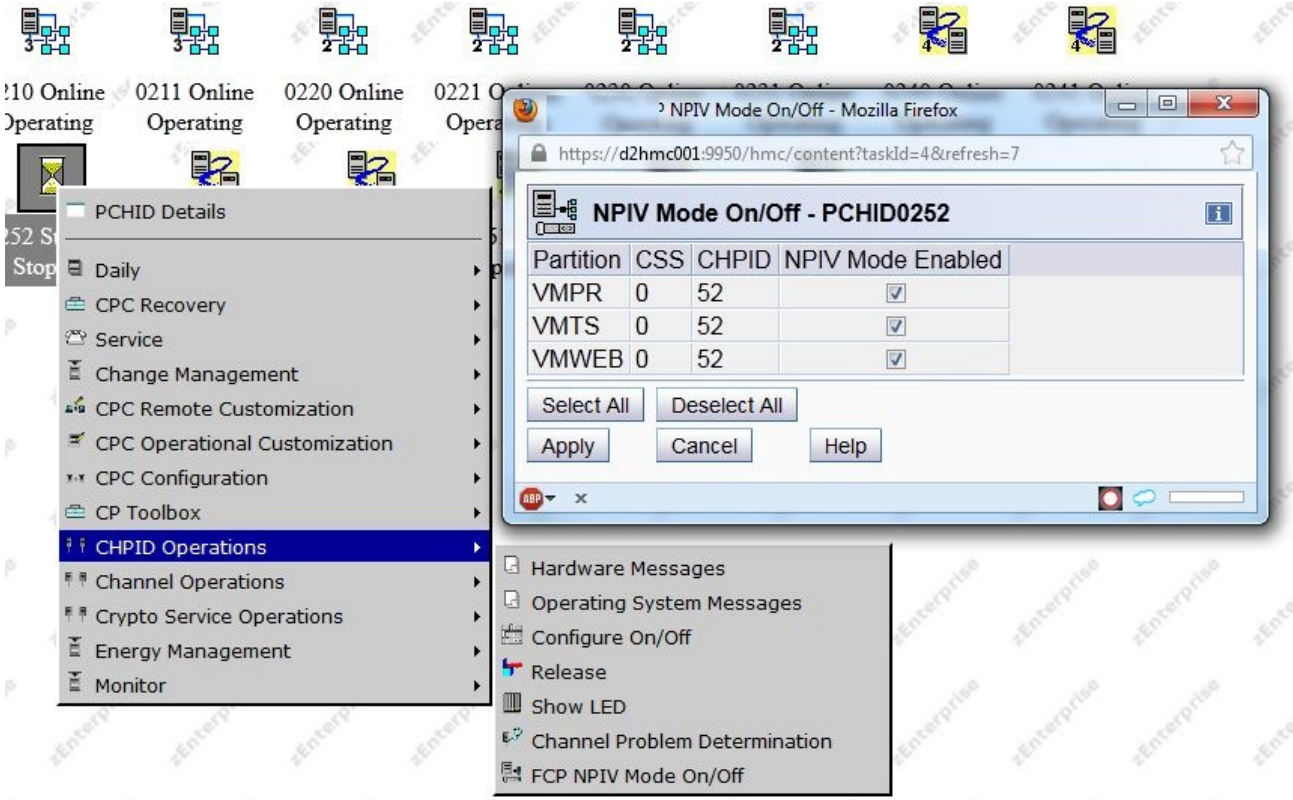
These settings give you 32 FCP addresses starting from 4000 (4000-401F) using PCHID 240.

Same configuration has been made for PCHIDs 241, 250 and 251.

In this case, you can set 32*4 FCP paths ergo you can have 32 hosts accessing your storage with 4 independent paths or 64 hosts with 2 paths etc.

Turn on NPIV

- NPIV allows multiple I/O devices to share one physical channel. Each I/O device has a unique WWPN which is visible to the SAN. It is an easier, more efficient and more secure method of assigning LUNS to your zLinux guests.

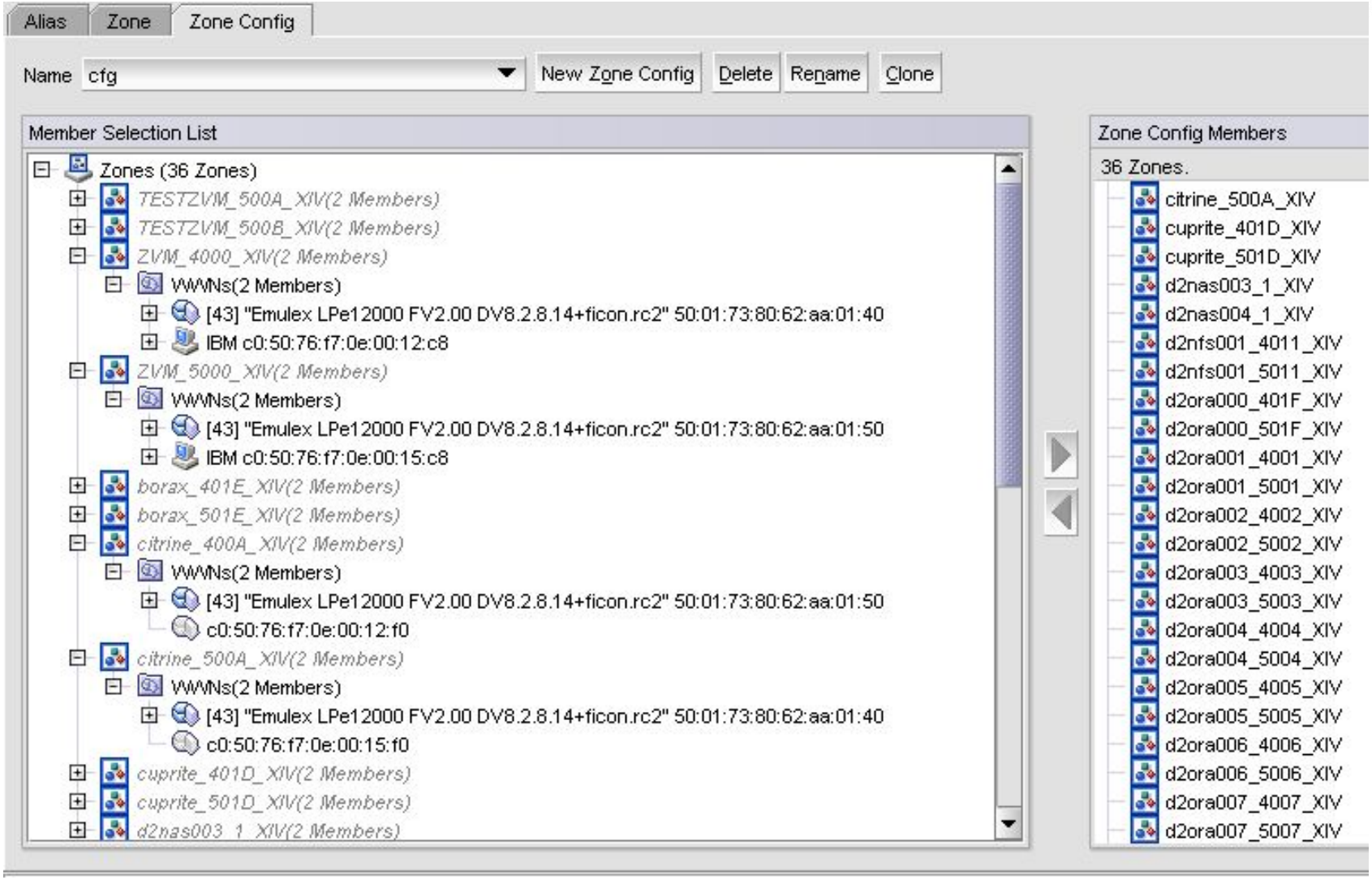


SAN Zoning

- Each zLinux (or z/VM) has 4 FCP devices (with multipathing), therefore 4 unique WWPNs (For example 4001, 4101, 5001,5101 – assigned in USER DIRECTORY).
- Each FCP device lives on a different PCHID (240,241,250,251 - defined in the IOCP profile)
- Two PCHIDs live on the first I/O card and two on the second I/O card.
- Each path (zLinux FCP device) -> (XIV FCP channel) has its own zone in one of SAN switches (defined in the SAN switch)

Each zLinux guest (and z/VM) has two zones in one switch and two zones in the other one. A lot of work at the beginning but after a while things seem to be clear and easy to manage.

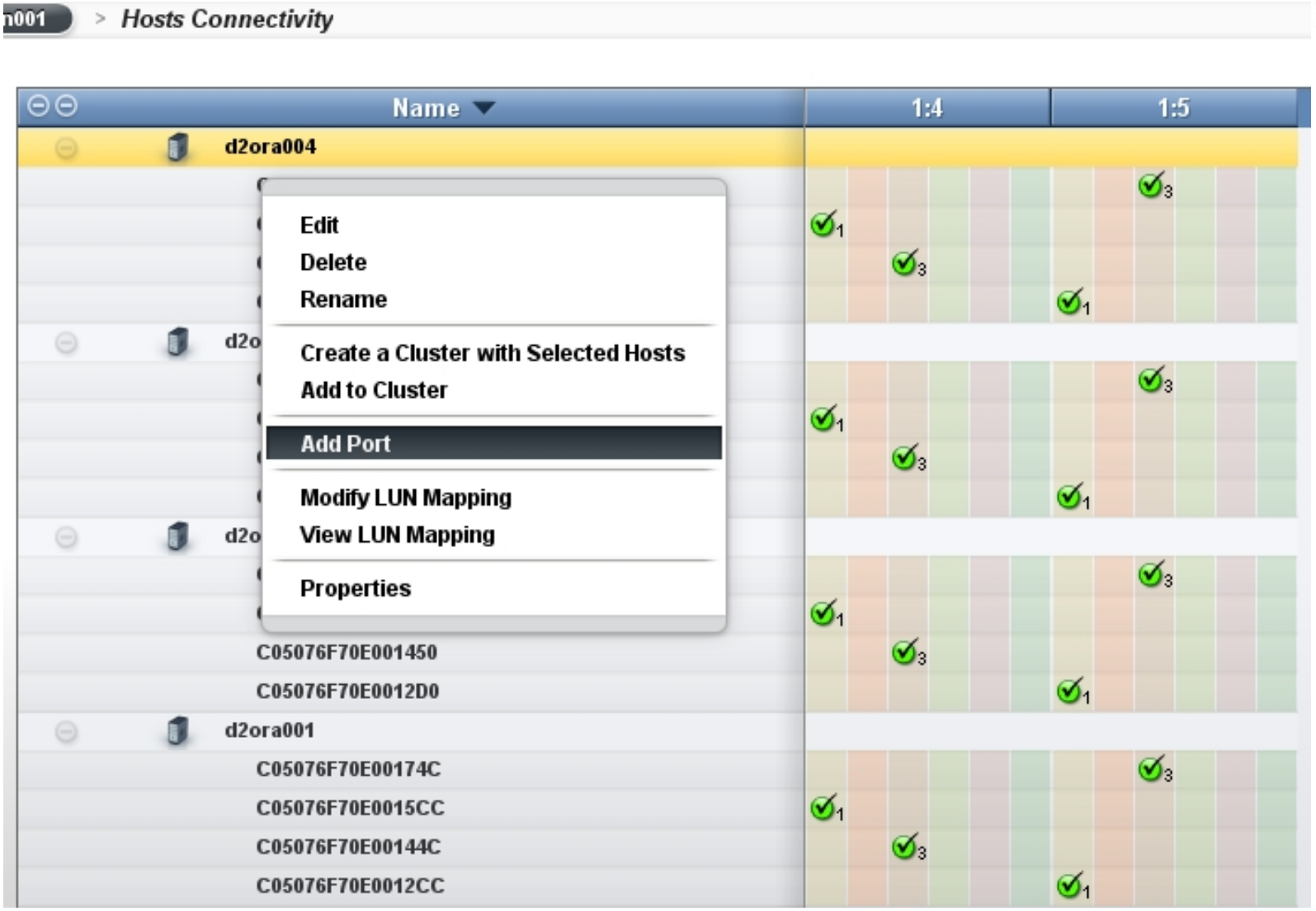
SAN Zoning



XIV – Host Connectivity

- Each zLinux and z/VM has its own host definition on XIV.
- Each defined host has a list of WWPNs (paths) which can access LUNs assigned for this host

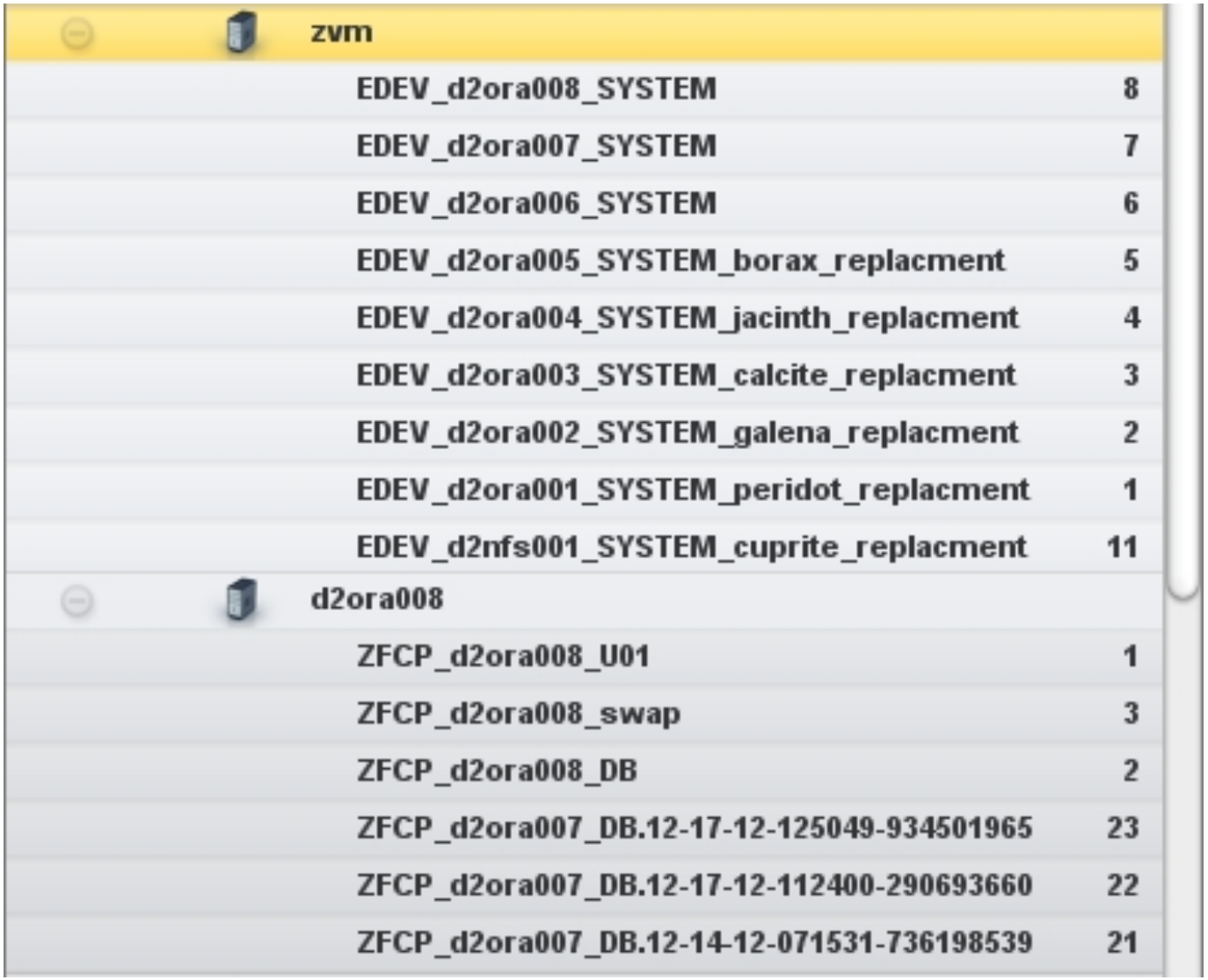
XIV – Host Connectivity



XIV – Volume mapping

- Each host definition has its own list of LUNs
  (Logical Unit Number)
- LUN slots are numbered 1-512 decimal
- Volumes are mapped to LUNs
- Linux and z/VM access LUNs (ergo disks) by hex number (need to convert)

XIV – Volumes by hosts



zvm	
EDEV_d2ora008_SYSTEM	8
EDEV_d2ora007_SYSTEM	7
EDEV_d2ora006_SYSTEM	6
EDEV_d2ora005_SYSTEM_borax_replacment	5
EDEV_d2ora004_SYSTEM_jacinth_replacment	4
EDEV_d2ora003_SYSTEM_calcite_replacment	3
EDEV_d2ora002_SYSTEM_galena_replacment	2
EDEV_d2ora001_SYSTEM_peridot_replacment	1
EDEV_d2nfs001_SYSTEM_cuprite_replacment	11
d2ora008	
ZFCP_d2ora008_U01	1
ZFCP_d2ora008_swap	3
ZFCP_d2ora008_DB	2
ZFCP_d2ora007_DB.12-17-12-125049-934501965	23
ZFCP_d2ora007_DB.12-17-12-112400-290693660	22
ZFCP_d2ora007_DB.12-14-12-071531-736198539	21

zLinux disks - EDEVICE and ZFCP

Options, pros and cons we have found:

- EDEVICE – emulated FBA over SCSI FCP device fragmented into minidisks (or fullpack) assigned to zlinux guests
  - CONS
    - Emulation impacts performance
    - If you clone it you have to change label before you use it
  - PROS
    - Only few FCP channels (ergo paths, zones) needed for all your EDEVICEs in system
    - All your tools (ddr, link, cpfmtxa etc) work fine
    - z/VM takes care for multipathing (not zlinux)

zLinux disks - EDEVICE and ZFCP

- ZFCP - FCP channel attached to a host therefore whole SCSI FCP device – linux takes care of SCSI subsystem
  - CONS
    - Each guest uses at least one FCP channel (more if multipathed)
    - Cannot link from one virtual machine to another
    - Each guest needs its own ZONEs in SAN and host definition on XIV
    - Clones have different World Wide Identifiers (WWID) than original – breaks multipath or fstab therefore booting process fails
  - PROS
    - Great performance

zLinux disks - EDEVICE and ZFCP

- Our choice:
  - EDEVICE for system disks (usually there aren't many IOPS here)
  - ZFCP for everything else (and swap)!

If you don't have to IPL from it, ZFCP it's a great choice

- Linux multipath daemon works great and has lots of options
- You can clone it or take a snapshot then mount it RW without even touching z/VM

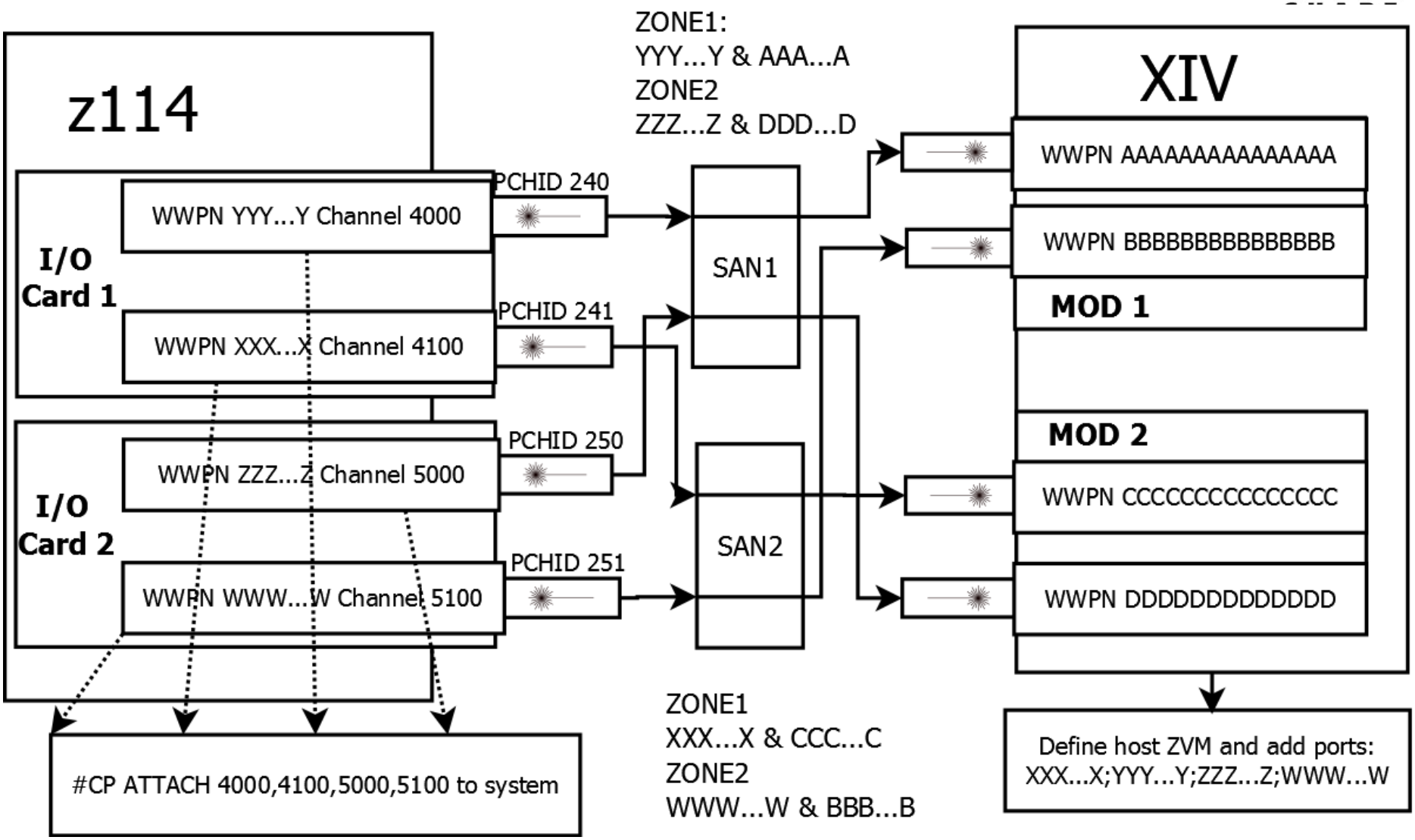
EDEVICE with multipath – step by step

One time steps:

1. Pick 4 FCP channels (for multipathing) in your z/VM and attach these to the system or leave as “FREE” (z/VM will attach these by itself during first EDEV creation). Use “QUERY FCP ALL” to find out which are available
2. Define 4 zones in your SAN, each zone should be a pair:
   1. WWPN of z/VM FCP channel (QUERY FCP – to find out what WWPNs are)
   2. WWPN of destination port in your XIV
3. Define ZVM host on XIV and add 4 ports to it

   (provide 4 WWPNs of the FCP channels you picked in step 1)

EDEVICE with multipath – step by step - example



EDEVICE with multipath – step by step

Steps 1,2 and 3 from above are only done once

Steps to be repeated for every new EDEV

4. Create Volume on XIV and map it to ZVM host (for example as LUN=9)
5. Create EDEVICE with address E009 in z/VM which points to LUN=9 by:

```
/* rexx */
'CP SET EDEV E009 TYPE FBA ATTR XIV',
'FCP_DEV 4000 WWPN AAAAAAAAAAA LUN 0009000000000000‘
‘VARY ONLINE E009’
 'CP SET EDEV E009 TYPE FBA ATTR XIV ADD PATH’,
'FCP_DEV 5100 WWPN  BBBBBBBBBBB LUN 0009000000000000',
'FCP_DEV 4100 WWPN  CCCCCCCCCCC LUN 0009000000000000‘,
‘FCP_DEV 5000 WWPN  DDDDDDDDDDD LUN 0009000000000000'
```

EDEVICE with multipath – step by step

In my example, ‘4000,5100,4100,5000’ are addresses of FCP channels which were picked in step 1.

AAA…A’–’DDD…D’ are target WWPNs – each one is paired with one of four FCP channels (4000-5000) in zone configuration.

‘000900000000000’ at the end is LUN number (HEX). So LUN 11 on XIV would be 000B000000000000.

EDEVICE with multipath – step by step

Check results:

```
#cp q edev e009 details
EDEV E001 TYPE FBA ATTRIBUTES XIV
  VENDOR: IBM PRODUCT: 2810XIV REVISION: 0000
  BLOCKSIZE: 512 NUMBER OF BLOCKS: 33609728
  PATHS:
    FCP_DEV: 4000 WWPN: AAAAAAAAAAA LUN: 0009000000000000
      CONNECTION TYPE: SWITCHED
    FCP_DEV: 5100 WWPN: BBBBBBBBBBB LUN: 0009000000000000
      CONNECTION TYPE: SWITCHED
    FCP_DEV: 4100 WWPN: CCCCCCCCCCC LUN: 0009000000000000
      CONNECTION TYPE: SWITCHED
    FCP_DEV: 5000 WWPN: DDDDDDDDDDD LUN: 0009000000000000
      CONNECTION TYPE: SWITCHED
```

EDEVICE with multipath – step by step

6. Depending on how you want to use it you can:
   - Attach E009 to yourself (maint e.g.) and label it with cpfmtxa or ickdsf, attach to system then define minidisks in the USER DIRECTORY in a similar way which you do with 3390s.
   - Attach E009 to linux guest and let it use whole device

For example:

MDISK ED01 FB-512 32 300000 E009

MDISK ED02 FB-512 300032 12000000 E009

MDISK ED03 FB-512 12300032 21309670 E009

Leave the first 32 blocks intact for the same reason as with the first cylinder for DASDs - These include volume label and allocation map.

EDEVICE with multipath – step by step

If you use Dirmaint, don't forget to add the new E009 to your extent file

```
*RegionId       VolSer   RegStart        RegEnd          Dev-Type
  E009          E009     32              33609720        9336-10
*Groups
LINLUNS   E009
```

7. Minidisks are ready to use by zLinux guest.

ZFCP with multipath – step by step

One time steps (for each new host):

1. Pick 4 FCP channels (for multipathing) in your z/VM and attach these to your linux host (or dedicate in DIRECTORY)

   Use "#CP QUERY FCP ALL" to find out which are available

   (preferably attach/dedicate real address AS different vdev)
2. Define 4 zones in your SAN, each zone should be a pair:
   1. WWPN of z/VM FCP channel (QUERY FCP – to find out what WWPNs are)
   2. WWPN of destination port in your XIV
3. Define Linux host on XIV and add 4 ports to it

   (provide 4 WWPNs of FCP channels which you picked in step 1)

ZFCP with multipath – step by step

Steps to be repeated for every new FCP disk for this host

4. Create Volume on XIV and map it to Linux host (for example as LUN=33 -> HEX=21)
5. If FCP channels are attached to your zLinux and channels are ONLINE then YAST should find new disk in ZFCP configuration (for SLES only)

   You should see four paths, one for each **channel**.

   These paths are pointing to the same **LUN**.

```
# lszfcp -PD | grep "0021"
0.0.1001/0xAAAAAAAAAAAAAA/0x0021000000000000 0:0:0:33
0.0.1002/0xBBBBBBBBBBBBBB/0x0021000000000000 1:0:0:33
0.0.1003/0xCCCCCCCCCCCCCC/0x0021000000000000 2:0:0:33
0.0.1004/0xDDDDDDDDDDDDDD/0x0021000000000000 3:0:0:33
```

ZFCP with multipath – step by step

6. Instead of using yast, you can configure your paths by using the command line (SLES example)

```
/sbin/zfcp_disk_configure '0.0.1001' '0xAAAAAAAAAAAAA' \
'0x002100000000000' '1'
/sbin/zfcp_disk_configure '0.0.1002' '0xBBBBBBBBBBBB' \
'0x002100000000000' '1'
/sbin/zfcp_disk_configure '0.0.1003' '0xCCCCCCCCCCCC‘ \
'0x002100000000000' '1'
/sbin/zfcp_disk_configure '0.0.1004' '0xDDDDDDDDDDDD‘ \
'0x002100000000000' '1‘
/sbin/mkinitrd && /sbin/zipl
```

‘1’ at the end of every line means “turn on”. If you wish to remove path use “0”

ZFCP with multipath – step by step

7. Four paths give you four disks (which are same disk in reality)

```
# lsscsi   | grep 33
[0:0:0:33]   disk    IBM      2810XIV          0000  /dev/sdm
[1:0:0:33]   disk    IBM      2810XIV          0000  /dev/sdn
[2:0:0:33]   disk    IBM      2810XIV          0000  /dev/sdo
[3:0:0:33]   disk    IBM      2810XIV          0000  /dev/sdp
```

8. Turn on multipath daemon

```
chkconfig multipathd on
chkconfig boot.multipath on
/etc/init.d/boot.multipath start
/etc/init.d/multipathd start
```

ZFCP with multipath – step by step

9. Define new alias in /etc/multipath.conf if you don't want to use long wwid based names

```
multipath      {
    multipath {
        wwid                    20017480062aa003d
        alias                   mpvol3
                }   }
```

10. Reload multipath devmap *(# multipath –r)*

ZFCP with multipath – step by step

11. Check results

```
# multipath -ll mpvol3
mpvol3 (20017380062aa003d) dm-2 IBM,2810XIV
size=1.9G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 0:0:0:3     sdc   8:32  active ready running
  |- 1:0:0:3     sdf   8:80  active ready running
  |- 2:0:0:3     sdi   8:128 active ready running
  `- 3:0:0:3     sdl   8:176 active ready running
```

12. Use ‘mpvol3’ alias with standard disk tools such as fdisk, parted, mkfs, mount etc.

Edevice & ZFCP

Keeping track of channels, disks, paths, labels etc is complex so we have few rules which make it a little bit easier.

1. All linux hosts have exactly the same mdisks layout with same sizes and similar lun map
   - 3 MDISKS for /boot ; /var ; / (from golden image)
   - LUN1 for swap
   - LUN2 for oracle software
   - LUN3 for data
2. Every linux virtual machine always has 4 dedicated FCP channels with the same VDEVs (1001,1002,1003,1004).
3. For all our EDEVICEs, VOLID=RDEV=E0XX where XX is used in virtual machine's name, XX is a LUN number on XIV and XX are last to digits of FCP channels dedicated for this Virtual Machine

Sample configuration

```
USER D2ORA006 XXXXXXX 4G 16G G
  INCLUDE LNXDFLT
  DEDICATE 1001 4006
  DEDICATE 1002 4106
  DEDICATE 1003 5006
  DEDICATE 1004 5106
  MDISK ED01 FB-512 36 300000 E006              * /boot
  MDISK ED02 FB-512 300036 12000000 E006        * /var
  MDISK ED03 FB-512 12300036 21309670 E006      * /
-------------------------------------------------------------------
q dasd details E006
E006  CUTYPE = 6310-80, DEVTYPE = 9336-10, VOLSER = E006
-------------------------------------------------------------------
(XCLI) d2xiv001>>mapping_list host=zvm
LUN   Volume                          Size   Master   Serial Number   Locked
….
6     EDEV_d2ora006_SYSTEM              17         90          no
….
```

EDEVICE wizard

With rules like this it was easy to create REXX scripts/wizards which automatically sets the EDEVICE and updates the virtual machine's profile.

In the XIV GUI the user makes a clone from the linux golden image and maps it to the ZVM host as LUN 'NN'

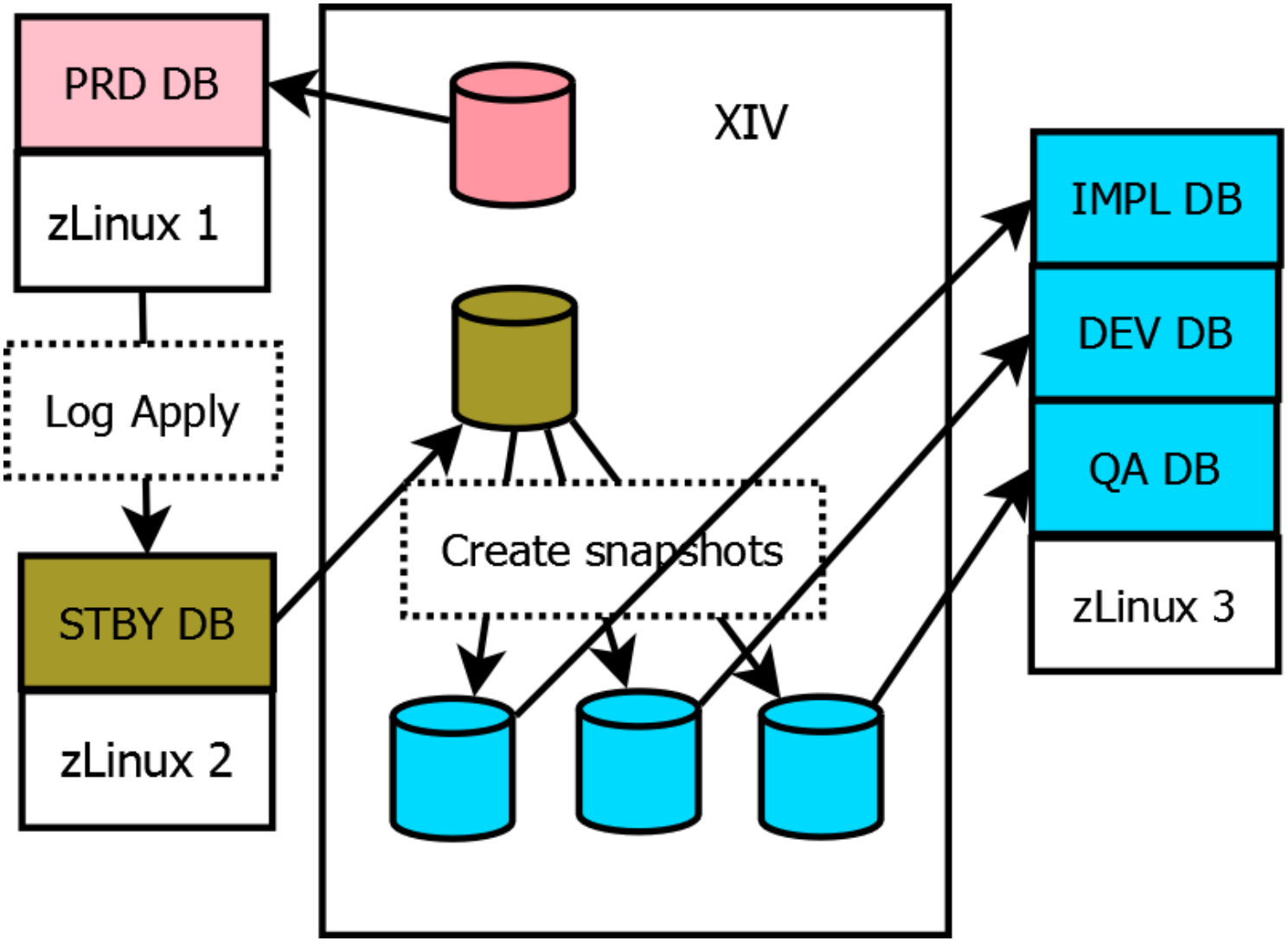
The user then runs the REXX wizard and types in the LUN number(NN) and the script automatically:

- sets EDEVICE with RDEV E0NN pointing to LUN 'NN'
- re-labels the new EDEVICE to E0NN
- updates the mdisks' definitions for E0NN
- attaches E0NN to the system
- updates autolog's profile exec or the system config

ZFCP – great for snapshoting! PRODSNAP

Using snapshot technology we have created tools which allow us to create--with one mouse click--a snapshot from our PROD database and use it for implementation, development, QA etc.

PRODSNAP



PRODSNAP – how it works?

- With a web gui, user picks a database he wants to “respawn” using PROD as a source
- Scripts does the whole work:
  - Shutdowns old DB
  - Unmounts old DB
  - Configures “off” FCP lun with old DB
  - Unmaps it from host definition in XIV configuration (with XCLI)
  - Deletes LUN with old DB on XIV
  - Stops log apply on Standby DB
  - Creates a snapshot from Standby DB
  - Starts log apply on Standby DB
  - Maps new snapshot to proper host definition in XIV config

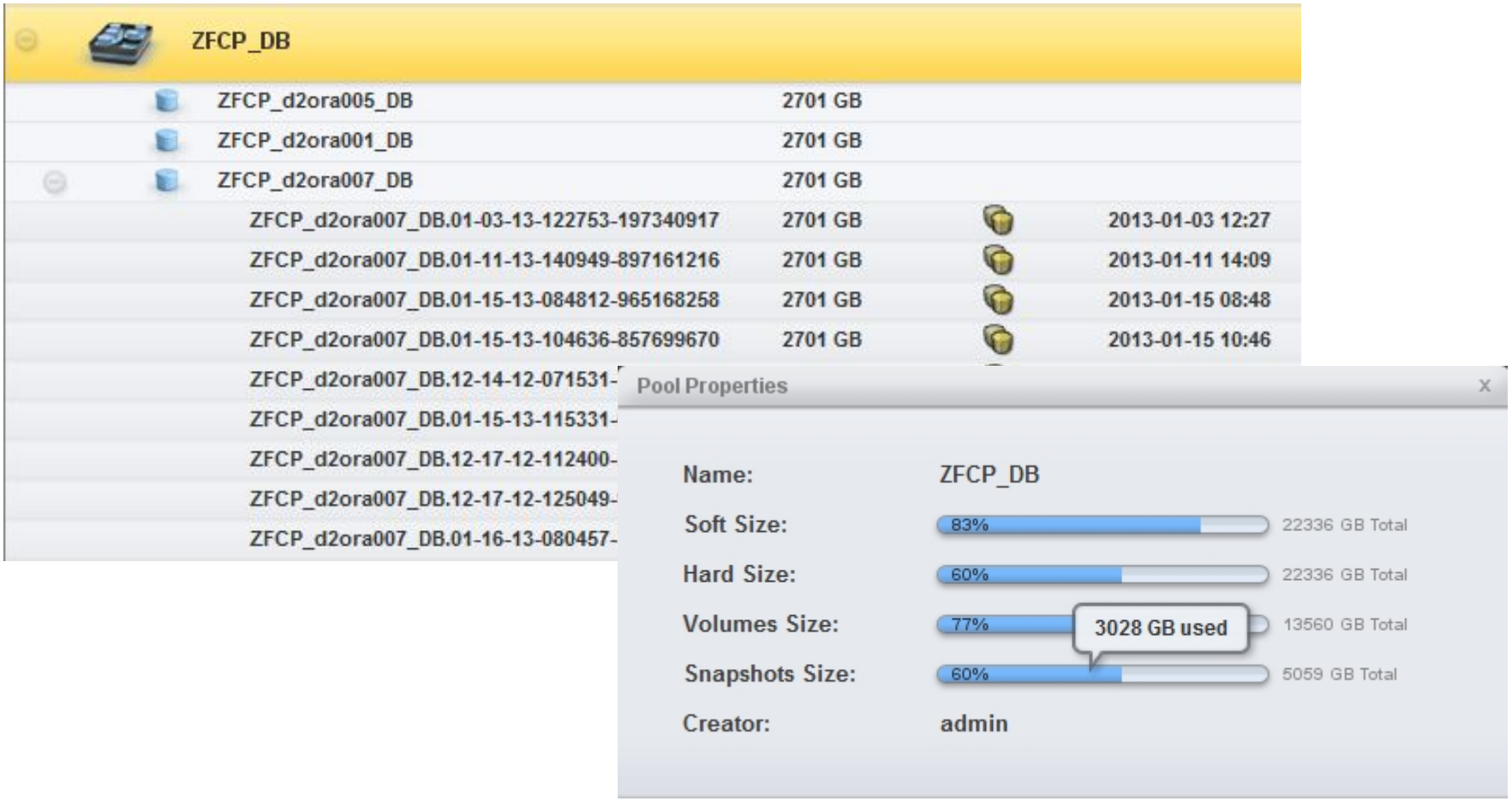
PRODSNAP – how it works?

- Unlocks snapshot for RW
- Configures “ON” a LUN which contains snapshot
- Mounts it in the proper directory
- Updates links (for oracle’s use)
- Runs some data scrubbing in database snapshot
- Starts new database snapshot

PROS:

- Developers and testers can work with data which is up to date and they can refresh it without engaging IT support.
- All these environments use much less space because they live as snapshots not real volumes.

PRODSNAP – how it works?



PRODSNAP – how it works?

There is a build process running (only 1 is allowed). Respawn is disabled currently. This page will automaticlly refresh every 5s

Databases on "d2ora008"

Slot	Name	mount	Lun serial	Serial Dec	Snap Name&Date	Status	
21	sqa1	/data21	20017380062aa0199	409	ZFCP_d2ora007_DB.12-14-12-071531-736198539	up	Respawn
22	sqa2	building...	building...	building...	shutting down database sqa2	up	Respawn
23	sqa3	/data23	20017380062aa019f	415	ZFCP_d2ora007_DB.12-17-12-125049-934501965	up	Respawn

Databases on "d2ora004"

Slot	Name	mount	Lun serial	Serial Dec	Snap Name&Date	Status	
24	demo1	/data24	20017380062aa01b9	441	ZFCP_d2ora007_DB.01-15-13-115331-608309863	up	Respawn

PRODSNAP – how it works?

```
Mon Dec 17 11:23:24 MST 2012 Creating sqa2 on d2ora008 for user gpo
Adding new env......
shutting down database sqa2
databese shutdown sucessfull
/dev/mapper/20017380062aa019b_part1 on /data22 type ext3 (rw)
new_env: Unmounting /data22
Configuring FCP disk 5001743062aa0150:0016000000000000
Configuring FCP disk 5001743062aa0142:0016000000000000
Configuring FCP disk 5001743062aa0140:0016000000000000
Configuring FCP disk 5001743062aa0152:0016000000000000
unmap_lun.sh:umapping lun=22 name=ZFCP_d2ora007_DB.12-14-12-102459-165151488
Command executed successfully.
del_snap: deleting ZFCP_d2ora007_DB.12-14-12-102459-165151488
Command executed successfully.
stopping log apply for mcopy
log apply stopped sucesfully
creating new snapshot...
Command executed successfully.
new_env.sh snapshot 20017380062aa019e ZFCP_d2ora007_DB.12-17-12-112400-290693660 created....
starting log apply for mcopy
log apply started sucesfully
map_lun.sh:maping ZFCP_d2ora007_DB.12-17-12-112400-290693660 as lun=22
Command executed successfully.
new_env.sh: unlocking new snapshot
Command executed successfully.
Configuring FCP disk 5001748062aa0150:0016000000000000
Configuring FCP disk 5001748062aa0142:0016000000000000
Configuring FCP disk 5001748062aa0140:0016000000000000
Configuring FCP disk 5001748062aa0152:0016000000000000
new_env.sh: mounting ZFCP_d2ora007_DB.12-17-12-112400-290693660 as data22.....
updating links....
Configuring database...make yourself a coffe
new database started sucesfully
Env sqa2 created sucesfulluy
-----------------------------------------------------------------------
```

Thin provisioning

Great idea, however...

- The filesystem has to be aware of thin provisioning capabilities in order to efficiently (quickly) reclaim unused space. IBM suggests using the Veritas File System (VxFS) by Symantec...which is not available for “Linux on system Z” distributions.
- There is a background reclaiming process in XIV which searches for “zeros”. When it finds a long sequence of zeros, it reclaims this space. It takes ages. Besides that, not every filesystem will “zero” space after deleting a file.

Summary

- Multipathing makes things more complex but more reliable
- NPIV gives you more flexibility and enhances z/vm virtualization
- Every path needs its own zone in the SAN switch
- EDEVICES are great for Linux system disks
- zFCP is good for data filesystems
- Snapshots are great
- Thin provisioning can be a great technology, however from our experience it turned out to be unusable in our environment

#SHAREorg

SHARE
Technology · Connections · Results

Running Linux virtual servers on your XIV under z/VM

Gregory Powiedziuk
Transzap, Inc.

4:30 PM on Monday, Feb 4, 2013
Session 12955

http://www.linkedin.com/pub/grzegorz-powiedziuk/6/137/11a/

SHARE in San Francisco 2013